EDUCATIONAL MANAGEMENT : TIMETABLE SCHEDULING ALGORITHMS

By SARATH B. SIYAMBALAPITIYA
Department of Engineering Mathematics, Faculty of Engineering,
University of Peradeniya, Peradeniya, Sri Lanka.

ABSTRACT

There is no doubt that the most important activity in schools and universities is conducting classes and lectures to their students. Preparation of timetables is a major task confronted by the management every year. Lot of time is spent on this exercise, yet the end result would be far from satisfactory. In this paper, we draw attention to the different approaches of solution to this problem and the need to look for new directions of research.

1. Introduction

Timetable scheduling is a problem faced by the management of educational institutions all over the world. Universities and schools are confronted with the timetable scheduling every year or semester. In many educational institutions timetable scheduling is carried out manually and in most cases a trial and error approach is adopted. Attempt have been made to develop computer software to handle these activities. However, the results are not very encouraging due to the inherent complexity of the problem. Even in institutions where computer software is used for the preparation of timetables a considerable part has to be done manually. It is a challenge to produce a sufficiently flexible timetable scheduling algorithm that could be applied for various higher educational institutions.

A timetable is a schedule of work arranged according to time and availability of resources. We need timetables to avoid conflicts, to optimize performance and also for minimizing costs. The goal of a scheduling algorithm should be to give a complete description of which teachers and students should meet, at which locations, and at what times. When evolving a timetable it is also necessary to construct it for the satisfaction of as many people as possible while respecting institutional traditions. There are different categories of teachers in a university like professors, lecturers, teaching assistants, instructors etc. for which teaching loads could be different. In a school, this distinction may not be visible very much. Rooms capacities, locations, and seating arrangements in rooms could differ. Students are grouped according to semesters or years or based on the elective or optional courses they offer.

Most of the available algorithms do not incorporate all these factors together. For instance, one may consider assigning the various subjects to different time slots without taking into consideration, the availability of lecture rooms or the capacity to accommodate the number of students taking the subjects or the preferences of lecturers. Therefore, the usefulness of these algorithms will be limited. However, most of the commercially available software for timetable scheduling are based on this type of algorithms.

One of the difficulties of this problem is the inability to device a general solution procedure. This is because of the inherent complexity of the problem as well as the large amount of data required to test problems of real size. There are different types of common scheduling problems in educational institutions. Examination timetabling occurs at schools or universities. Course timetabling exists in universities and class-teacher timetabling at schools. Each category has its own characteristics. For example, in examination timetable scheduling, sufficient time intervals should be allowed between consecutive examinations of the same student. It deals with courses,

students, and timeslots only . Here, it is required to assign each course to a timeslot such that no student has to be present at two or more examinations simultaneously. In the class-teacher assignment problem, set of teacher-course meetings already assigned to a set of timeslots is given and it is required to place these into a set of available class rooms. At universities, due to the presence of large number of different programmes, it is extremly difficult to device a conflict free timetable. Therefore, the goal would be to try to find the timetable with the least number of conflicts.

The timetable scheduling problem in whatever form is known to be a NP complete problem [Karp,1972] for which no polynomial time algorithms could be developed. Many researchers have worked on finding approximate solutions to the problem. However, it is difficult to make comparisons among different approaches mainly because they have proposed solution methods for highly localized problems. Formulation of the problem and the data used could not be applied for a general situation applicable to different types of Institutions.

A common approach to solve this problem is by using heuristic algorithms. In one heuristic algorithm [Barham and Westwood, 1978] optional subjects were scheduled so that every student could attend as many courses as possible. Students were allowed to change topics if clashes appeared in the timetable. To minimize clashes it was suggested to group the optional subjects before constructing the timetable.

Another heuristic approach [Willemen,2002] consists of a tree search algorithm that extends the subject group assignment step by step, and a tabu search algorithm that repairs the current timetable if the subject group assignment introduces resource conflicts. The tabu search method is integrated in the tree search algorithm. That is, the main step of this algorithm first changes the subject group assignment and then updates the time slot assignment. In this manner, a feasible complete timetable is maintained in each step.

The goal of timetable scheduling could be stated as [White and Zhang ,1998]: An automatic timetabling system should formulate complete descriptions of which students and which teachers should meet , at what locations , at what times and should accomplish this quickly and cheaply while respecting the traditions of the institution and pleasing most of the people involved most of the time. However, it is easier said than done. They have described two approaches to solve the timetable scheduling problem the constraint logic approach and tabu search.

2. Constraints

The majority of the educational institutions do not have enough resources to satisfy all the requirements. Therefore, it is necessary to impose conditions which will be formulated as constraints in the mathematical model. The constraints for the timetable scheduling problem can be divided into two types, primary constraints and secondary constraints. Primary constraints are hard constraints which have to be satisfied strictly. However, the secondary constraints are soft constraints which could be relaxed if necessary. Usually, the secondary constraints involve desirable features which may not be compulsory for the successful implementation of a timetable.

Some of the primary constraints could be:

Only one course conducted by a teacher can be scheduled at the same time slot.

(It is not possible for a teacher or a student to be present at more than one lesson simultaneously)

No two courses taken by the same group of students can be scheduled simultaneously.

Each class room can be used for only one lesson during

a time slot.

Class rooms assigned must have sufficient capacities to accommodate the relevant group of students.

Certain courses should be scheduled simultaneously but in different class rooms (eg. smaller tutorial groups for the same course).

Core courses and optional courses should be scheduled separately without overlapping.

Provision should be made to schedule each course simultaneously offered by students in different semesters.

It must be possible to assign two or more time slots consecutively on certain days for a particular course (eg. Laboratory classes will need this arrangement).

Some of the secondary constraints could be stated as:

A teacher should have atleast one time slot free between any two teaching time slots.

Teacher should have atleast one time slot free between any two teaching time slots.

Friday afternoon should be made free as far as possible.

Courses should be scheduled into classrooms in the buildings as close as possible (This will minimize the travel time between classes).

Lessons of the same subject should be assigned to different days of the week. (This will help the students to have sufficent time to absorb the material taught and to be in a position to do homework and submit assignments in time).

Recommended optional courses for a student should be scheduled in such a way to have the fewest time conflicts among them.

The major components of the scheduling process involves teachers or lecturers, time allocations, class rooms and courses. Some of the desirable features of a successful timetable is that if lecturers conduct several classes on a particular day , there must be a gap of atleast one hour (period) between consecutive classes. It may also not be suitable to schedule a lecture very early in the morning and the next lecture in the late afternoon for the same lecturer on the same day. On the other hand, the assignment of the lectures must be such that the teachers are able to spend sufficient time for preparation and carrying out research. It is ideal, if it is possible to give a full day per week free from classes for a teacher. They must also be allowed sufficient preparation time between lectures so that the lectures will be of high quality.

When we consider the student population, we may not be able to avoid consecutive hours of classes , as the total number of credits for each programme has to be covered fully in time. This may not be possible without making student timetables almost full on most of the days. However, it is desirable to allow some breathing space for students between classes wherever possible. Otherwise, the students will lose concentration and may not be able to fully understand the material covered in the lectures. Instead, they will tend to write down only the notes without paying much attention for understanding which will affect the overall performance of the students.

The number of students registered for each course could be different in general. Usually, the size of different class rooms will be different. Therefore, the need of assigning the courses for compatible class rooms will arise. One of the difficulties of scheduling in many instiutions is that classes have to be conducted from 8.00 am in the morning atleast upto 6.00 pm due to the large number of courses to be scheduled.

3. Objective Function

Timetable scheduling problem can be considered as a resource allocation problem where teachers, students,

class rooms, and time slots are allocated to lessons. A lesson could be interpreted as a lecture, tutorial, laboratory class etc. There are three straight forward decisions to make. They are the teacher assignment, room assignment and time slot assignment for each lesson. However, assigning students to lessons depends on the courses offered by the students.

Formulating a standard objective function for this problem has been found to be difficult. It is not possible to construct an objective function to be maximized or minimized which is acceptable to everybody. Therefore, the goal is to obtain a feasible solution satisfying all the constraints let alone finding an optimal solution. In many situations, even obtaining a feasible solution could become extremely difficult. Hence, many would be satisfied with a solution which will minimize the violation of the number of constraints. Once this is done, the remaining part of the scheduling problem have to be worked out manually. Although the objective is always to develop an acceptable timetable , the problems differ in the way acceptance is defined.

4. Mathematical Model

Mathematical models presented by the researchers also differ in the way the scheduling problem is defined. Here, we consider some approaches which takes into consideration, the important constraints we have already mentioned.

First we shall consider a class-teacher model. We assume that a class is defined as a group of students who follow the same programme.

Let C = $\{c_1, c_2, \ldots, c_m\}$ be a set of classes and T = $\{t_1, t_2, \ldots, t_n\}$ a set of teachers.

Let **R** $= (r_{ij})$ be the requirement matrix (of order mn) where r_{ij} is the number of lectures involving class c_i and teacher t_j.

Assume that the number of time periods is p (having equal length).

Now, the problem is to assign each lecture to some period so that no teacher or a class is involved in more than one lecture at a time.

Define

$$x_{ijk} = \begin{cases} 1 & if \ \ class \ \ c_i \ \ and \ \ teacher \ \ t_j \ \ meet \ \ at \ \ period \ \ k \\ 0 & otherwise \end{cases}$$
$$(i = 1, 2, \ldots, m; \;\; j = 1, 2, \ldots, n; \;\; k = 1, 2, \ldots, p)$$

We have to construct a schedule satisfying the following constraints.

$$\sum_{i=1}^{m} x_{ijk} \le 1 \quad (j = 1, 2, \ldots, n; \;\; k = 1, 2, \ldots, p)$$

(Specifies that each teacher-period has atmost one class at a time)

$$\sum_{j=1}^{n} x_{ijk} \le 1 \quad (i = 1, 2, \ldots, m; \;\; k = 1, 2, \ldots, p)$$

(specifies that each class-period has atmost one teacher involved)

$$\sum_{k=1}^{p} x_{ijk} = r_{ij} \quad (i = 1, 2, \ldots, m; \;\; j = 1, 2, \ldots, n)$$

(Requires that the number of meetings of class i and teacher j conforms to the specifications of the requirements matrix)

The course scheduling problem consists of assigning each lecture to some period (time slot) of the week in such a way that no student is required to take more than one lecture at a time.

Let m be the number of time periods , k = 1,2,…,m;

n_1 be the number of classes , i = 1,2,…,n_1

n_2 be the number of teachers j = 1,2,…,n_2

n_3 be the number of rooms , l = 1,2,…,n_3

Let **R** $= (r_{ij})$ be the requirement matrix (of order $n_1 \times n_2$)

where r_{ij} is the number of lectures involving class i and teacher j. Let **E** be an n_1-vector whose ith element is the enrollment in class i and **C** be an n_3-vector whose lth element is the capacity of room l.

Define

$$x_{ijkl} = \begin{cases} 1 & \text{if class } i \text{ and teacher } j \text{ meet at period } k \text{ in room } l \\ 0 & \text{otherwise} \end{cases}$$

We have to construct a schedule satisfying the following constraints

$$\sum_{i=1}^{n_1} x_{ijkl} \le 1 \quad (j = 1, 2, \ldots, n_2;\ k = 1, 2, \ldots, m;\ l = 1, 2, \ldots, n_3)$$

(specifies that each teacher-period-room has atmost one class in it)

$$\sum_{j=1}^{n_2} x_{ijkl} \le 1 \quad (i = 1, 2, \ldots, n_1;\ k = 1, 2, \ldots, m;\ l = 1, 2, \ldots, n_3)$$

(specifies that each period-class-room has atmost one teacher involved)

$$\sum_{l=1}^{n_3} x_{ijkl} \le 1 \quad (i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2;\ k = 1, 2, \ldots, m)$$

(specifies that each teacher-period-class occupies at most one room)

$$\sum_{l=1}^{n_3} \sum_{k=1}^{m} x_{ijkl} = r_{ij} \quad (i = 1, 2, \ldots, n_1;\ j = 1, 2, \ldots, n_2)$$

(Requires that the number of meetings of class i and teacher j conforms to the specifications of the requirements matrix)

$$\sum_{j=1}^{n_2} \sum_{k=1}^{m} e_i x_{ijkl} \le \sum_{j=1}^{n_2} \sum_{k=1}^{m} C_i x_{ijkl} \quad (i = 1, 2, \ldots, n_1;\ l = 1, 2, \ldots, n_3)$$

(specifies that all rooms are large enough to contain the classes scheduled into them)

5. Solution Approaches

Different types of solution approaches have been attempted to solve the timetable scheduling problem. We briefly mention some of the more common solution approaches here.

One of the simplest approaches to construct a timetable is to use a greedy algorithm. Using this approach an attempt could be made to minimize the number of conflicts. In other words we try to minimize the number of violated constraints. The timetable is initially empty. It tries to build the timetable step by step by assigning students into classes by making the smallest number of conflicts. Usually faculty constraints are considered as high priority over the student preferences when scheduling the classes.

Graph coloring [de Werra,1985] is another approach that has been used to solve the timetable scheduling problem. The solutions to class-teacher scheduling and course scheduling problems have been obtained using this approach.

In the course scheduling problem, each lecture of each course is associated with a node. An edge connects two nodes if atleast one student is registered in the courses associated with each node.

Tabu search [Glover, 1989] is another approach used to solve the timetable scheduling problem. Tabu search is a heuristic procedure designed to avoid getting trapped in local optimal solutions. It keeps information on the itinerary through the previous solutions generated in the search for solutions. Such information will be used to guide the move from one solution to the next solution. This approach forbids certain moves in the neighbourhood and to guide search through more promising directions.

A special feature of the tabu search approach for timetable scheduling is that a penalty is imposed for the violation of each constraint. This approach tries to minimize the constraint violations by considering different possible assignments. First, an initial version of the timetable is constructed which satisfies all the primary constraints. The value of the penalty for this timetable is computed at this stage. Algorithm proceeds by applying a simple perturbation to the current time table. For

example, class-teacher pairs exchange their time slots and class rooms. In this manner, a feasible perturbation leads to neighboring timetables. Heavy penalties are imposed on violation of more important constraints while the violation of less important constraints are given light penalties.

Another approach used for timetable scheduling is logic programming [Le Kang,et.al. 1991]. Logic programming is a style that, like functional programming , is based on writing facts. Where in functional programming the facts are equations , in logic programming the facts are implications. The idea is to write a collection of facts, and then to let an inference engine work with those facts, deriving new facts from old. The algorithm which finds a solution to the timetable scheduling problem is a modification of the standard Prolog back tracking algorithm based on depth first search. The following sequence of steps is followed. Choose a time block in the specified time zone. Then choose a class room of the proper type. Check whether this time-classroom pair violates any constraints.

If no constraints are violated, the course will be scheduled into that classroom during that time block. If not, backtracking is performed with different class rooms and different time blocks. If no feasible solution is found after this process, the secondary constraints are relaxed group by group, and the algorithm is applied again.

An algorithm developed by a combination of constraint logic and tabu search [White and Zhang, 1998] was also used for the timetable scheduling. . Constraint logic is a form of logic programming in which the variables are subject to constraints.

Evolutionary algorithms has also been used to solve this problem [Burke and.Newall,1999]. In this approach, the scheduling problem has been decomposed into smaller components each of which is of a size that the evolutionary algorithm can effectively handle.

6. Conclusion

We have briefly discussed some of the solution approaches used to solve the timetable scheduling problem. Comparison between different approaches has been difficult due to the many different models considered in the literature. As a result, it is not possible to claim the superiority of one method over another in a general sense. In spite of the vast amount of research conducted, there appears to be no known method which could be applied to any general timetable scheduling problem successfully. A major reason for this situation could be the inherent complexity of the problem. The other reason is that most of the studies concentrated on highly localized timetable scheduling problems, which has little relevance to other institutions. Another drawback in existing methods is that they are unable to produce alternative timetables when the need arises. Even a slight change to an existing schedule may result in a timetable with increased amount of conflicts. The other disadvantage is that the necessity of manual adjustment even with the presence of software packages.

The need for a fully automated timetable scheduling system which could be applied across different institutions has been felt more than ever, in recent years. This is because of the increased number of courses and combinations offered, the timetable scheduling is becoming increasingly difficult in many institutions around the world. It has become necessary to look for new directions of research without being tied to traditional solution approaches.

References

(1) Barham,A.M. and Westwood,J.B.,[1978] *A Simple Heuristic to Facilitate Course Timetabling, Journal of the*

Operational Research Society, Vol.29, No. 11, pp. 1055-1060

(2) Burke, E.K., and Newall,J.P.,A [1999], *Multistage Evolutionary Algorithm for Timetable Problem, IEEE Transactions on Evolutionary Computation, Vol.3, No. 1, pp. 63-74.*

(3) De Werra, D.[1985], *An Introduction to Timetabling, European Journal of Operational Research, pp. 151-162.*

(4) Glover,F. [1989], *Tabu Search, Part 1, ORSA Journal on Computing, 1, pp. 190-206.*

(5) Karp,R.M.[1972] ,*Reducibility among Combinatorial Problems in Complexity of Computer Computations ,Plenum, New York,pp. 85-103.*

(6) Le Kang, Schoenberg, G.H.V.,and White,G.M.[1991],*Complete University Timetabling using Logic, Computer Educ. Vol. 17, No. 2, pp. 145-153.*

(7) White,G.M.and Zhang,J [1998].,*Generating Complete University Timetables by Combining Tabu Search with Constarint Logic,Lecture Notes in Computer Science,Vol. 1408.*

(8) Willemen, R.J [2002].,*School Timetable Construction : Algorithms and Complexity ,Ph.D. Thesis, Technische Universiteit Eindhoven, 2002.*

ABOUT THE AUTHOR

Prof. Sarath B.Siyambalapitiya is the Head of Department of Engineering Mathematics, Faculty of Engineering, University of Peradeniya, Sri Lanka. His research interest are Operations Research % Optimization , Engineering &Mathematics Education.

Sarath B. Siyambalapitiya